1.

Write a program which prints all factors of number N. The code which prints the factors should be placed inside a function. Print information if N is a prime number.

2.

Modify the program in such a way that it finds all prime numbers inside a range $\{1, \ldots, \mathrm{M}\}$. Modify function which prints factors so it only returns information that a number is a prime number or not.

3.

Write a program to see how you can use pointers and arrays:

- declare integer variable `d`,
- initialize `d` with `10`,
- declare integer pointer `p`,
- initialize pointer `p` with the address of the variable `d` (`p` points to `d`),
- print value of `d` and value which is pointed by `p` (use operator `*`),
- change value of `d` to `20`,
- print value of `d` and value which is pointed by `p`,
- change value pointed by `p`,
- print value of `d` and value which is pointed by `p`,
- print address of `d` and value of the pointer `p` (remember that pointer stores the address, use `printf` function, e.g.: `printf("%p", &d);`)
- declare array `tab` of integers with two elements (`int tab[2];`)
- initialize elements of the array with: `333` and `444`,
- assign to the pointer `p` address of the array `tab`,
- fill the following table:

address of the variable (L-value)	name of the variable	value of the variable (R-value)
%p	d	%d
%p	p	%p
%p	p[0]	%d
%p	t	%p
%p	t[0]	%d
%p	t[1]	%d

- find the distance (in bytes) between addresses of the first and the second element of the table `tab`